INTERACTIVE CLOUDS

LOAD BALANCING WITH DREAMFACE

GET STARTED BY DEFINING OBJECTIVES

Load Balancing is not just a feature, it is a requirement for an Enterprise class platform to run applications

Cloud Application Platform

ROBUSTNESS

Architecture designed to support multi-tenancy

Service Continuity

Reliability

Fail Over

manageable & configurable

MONITOR & ALERT

Manage concurrent users

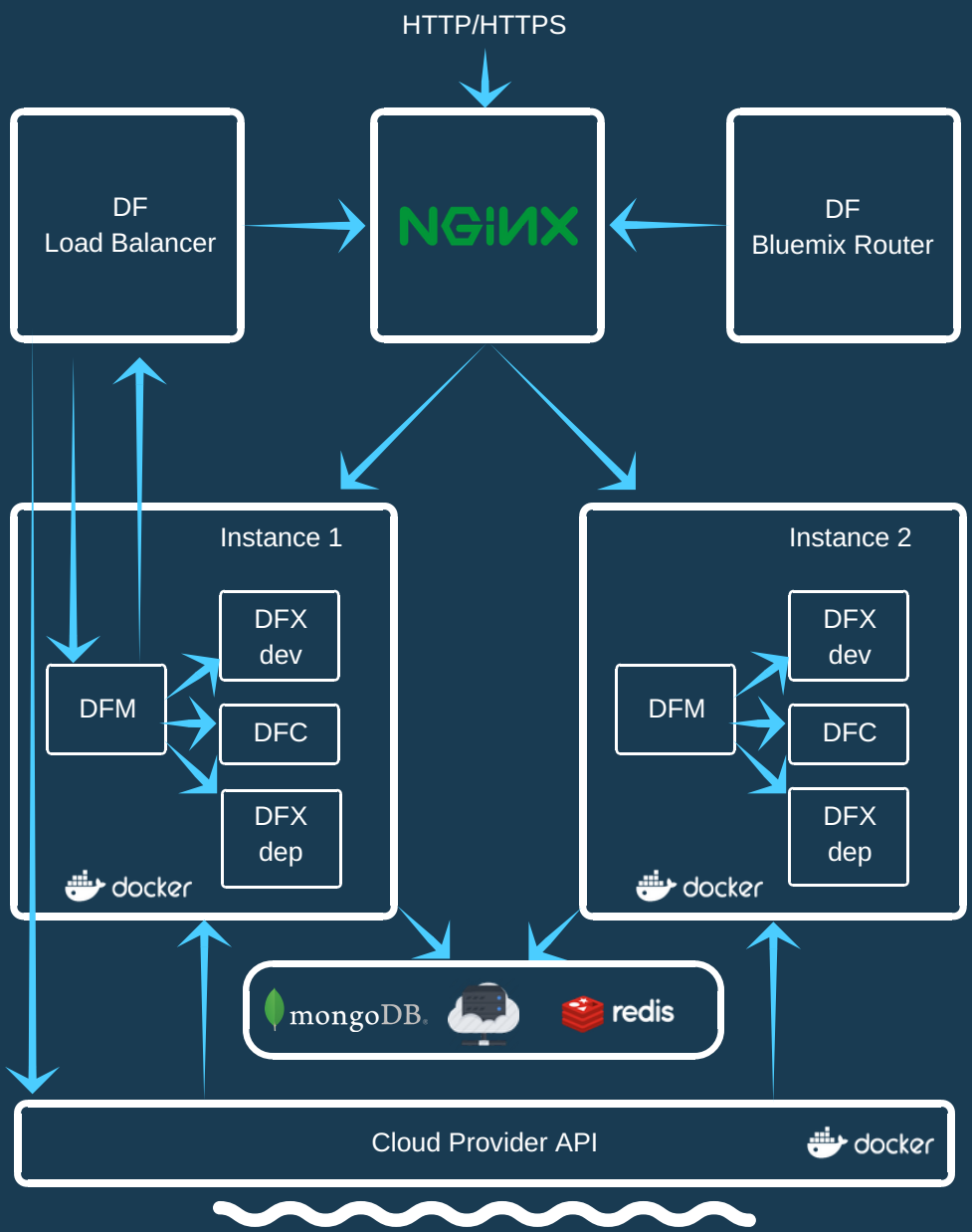
PERFORMANCES

Provide and benefit from an elastic architecture.

The Components

DFX	DreamFace X-Platform, declined into 2 editions: DreamFace for Development (dev) & DreamFace for Deployment (dep)
DFC	DreamFace Compiler: compiles and deploy applications
DFM	DreamFace Manager: Starts, stops, updates and monitor each component of an instance (DFX dev, DFX dep, DFC)
DFLB	DreamFace Load Balancer: Manage load balancing of requests
DFBR	DreamFace Bluemix Router: manage requests from Bluemix (create tenant, remove tenant, authenticate Bluemix user)

The Big Picture

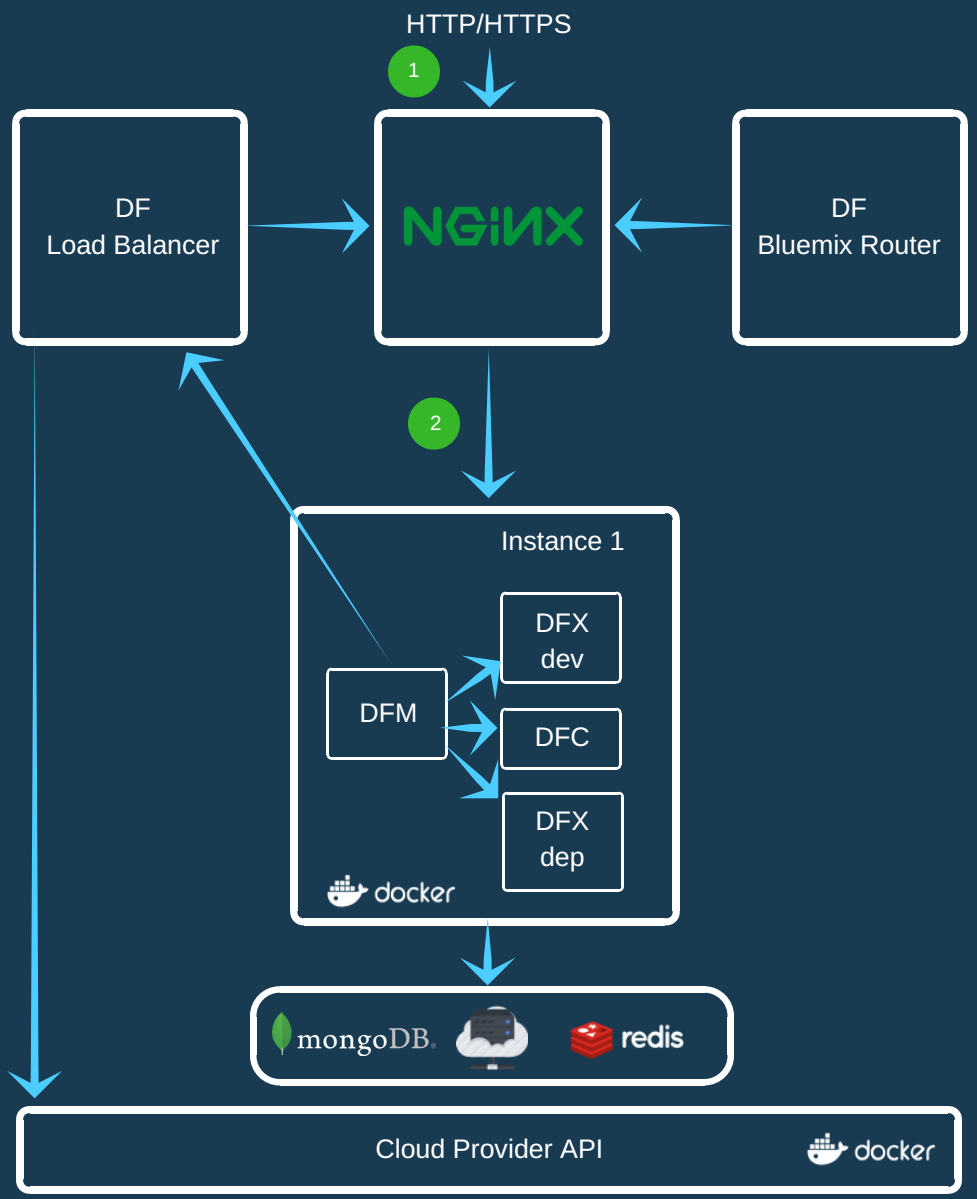


The Scenario

Normal Flow

01 Load Balancing

- (1) Some http requests are sent to NGINX from Internet
- (2) NGINX proxies all requests to Instance 1

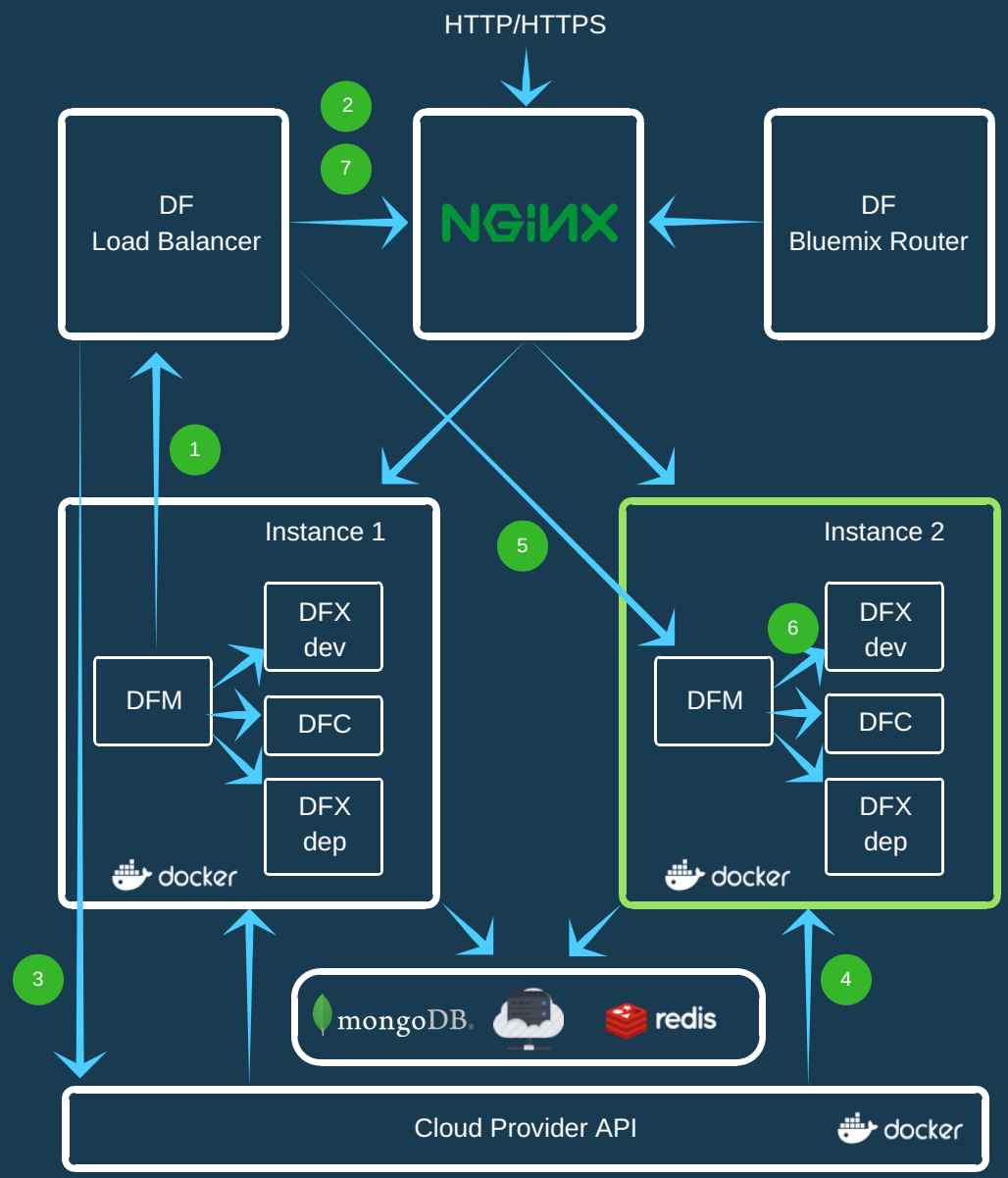


The Scenario

Instance Overload

02 Load Balancing

- (1) DFM (Instance 1) sends a notification to LB
- (2) DFLB asks NGINX for statistics and defines what tenants caused the overload
- (3) DFLB sends a request to Cloud Provider to create new instance
- (4) Cloud Provider creates a new instance (Instance 2) based on a pre-defined image
- (5) DFLB sends a request to DFM to start required components (ex: DFX dev), and its appropriate configuration (what tenants it should operate)
- (6) DFM starts the components, and responds to DFLB when it is completed
- (7) DFLB changes configuration of NGINX, so that it will proxy requests to both Instance 1 and Instance 2 depending of tenant ID

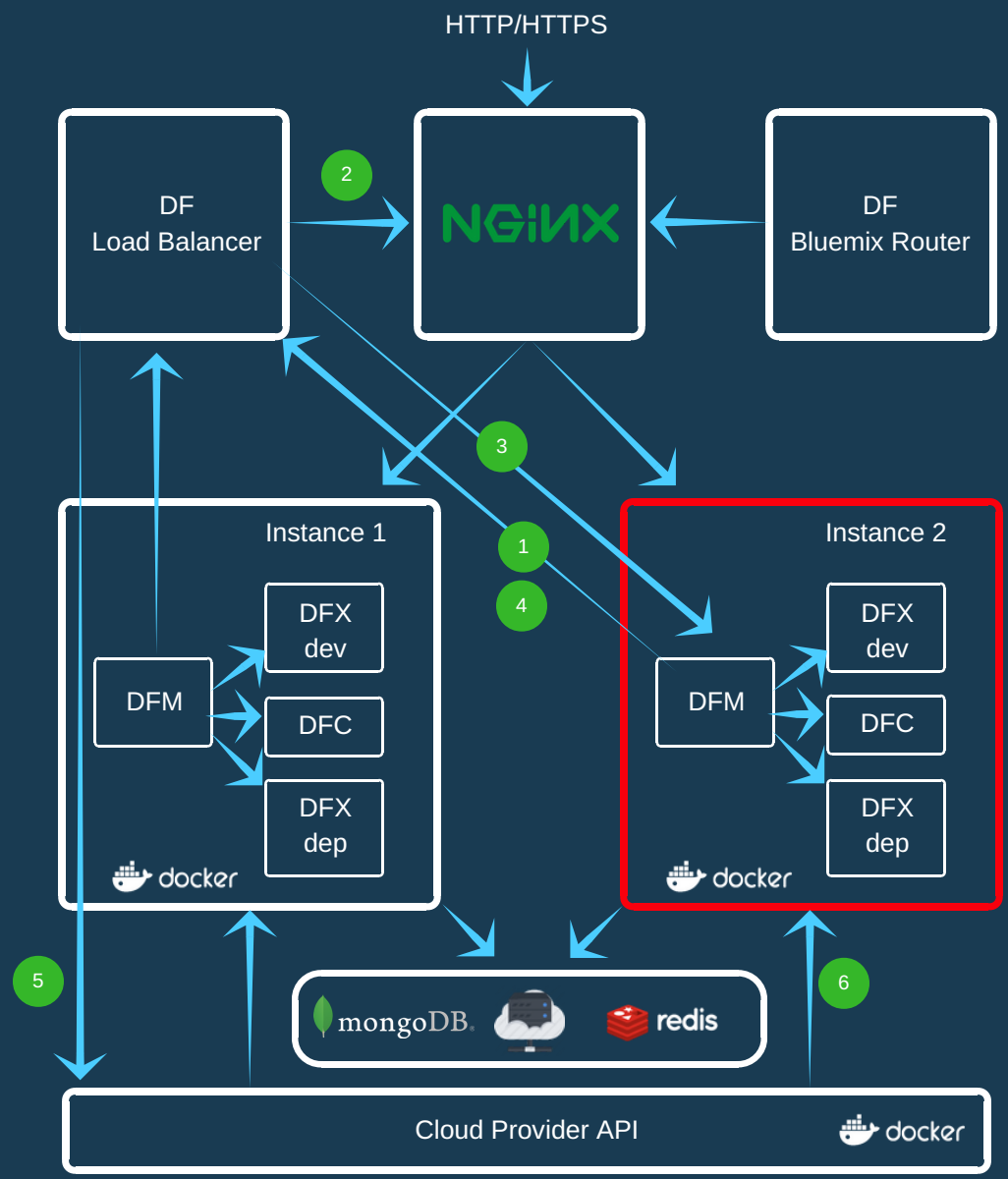


The Scenario

Low Loaded Instance

03 Load Balancing

- (1) DFM (instance 2) sends a notification to LB to inform about a low usage
- (2) DFLB changes NGINX configuration to not use Instance 2
- (3) DFLB sends a request to DFM (instance 2) for a graceful shutdown (all components are stopped, all pending requests terminated)
- (4) DFM notifies DFLB that all components are stopped
- (5) DFLB sends a request to Cloud Provider to remove instance 2
- (6) Cloud Provider removes the instance 2

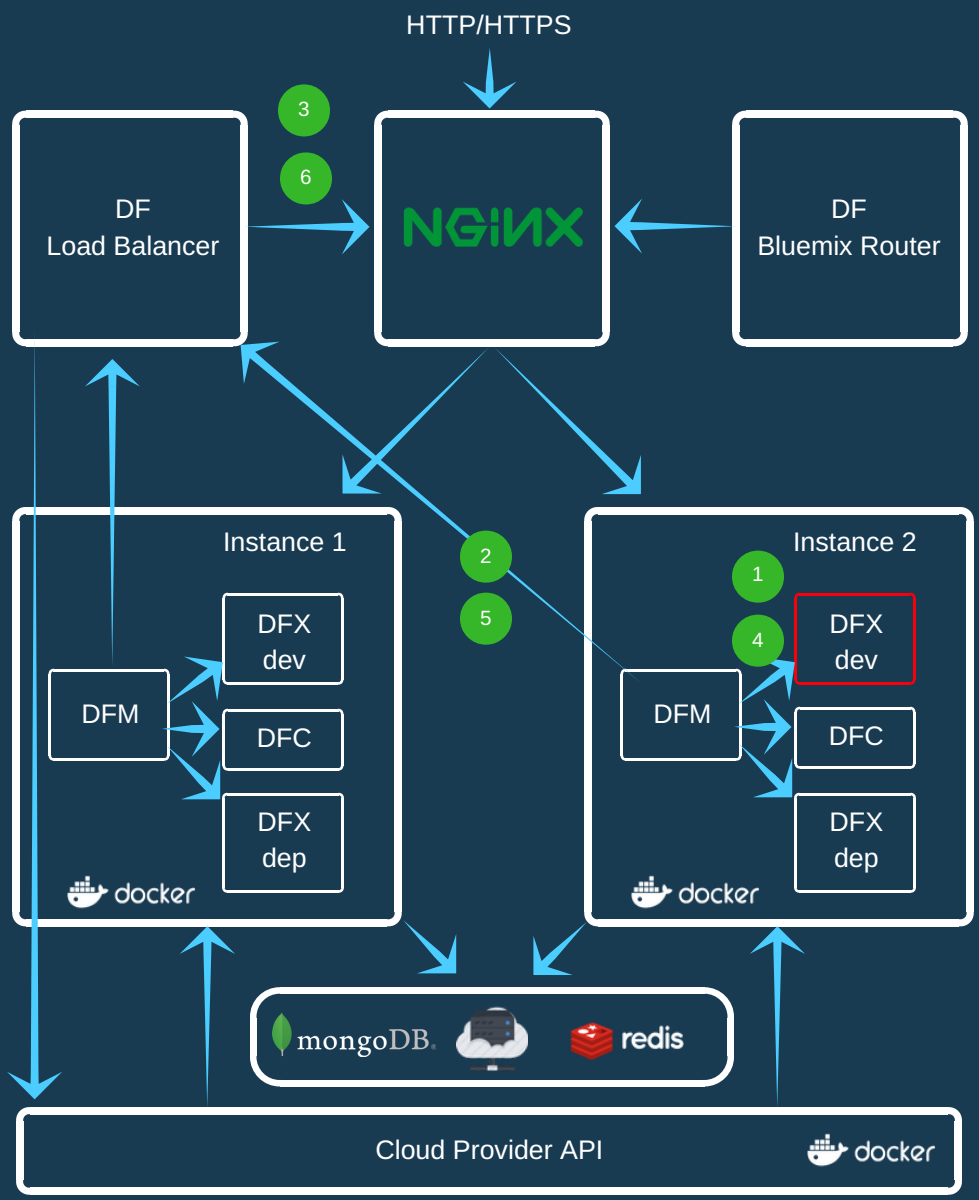


The Scenario

Broken component

04 Load Balancing

- (1) DFM (instance 2) detects a component is down
- (2) DFM sends a notification to LB
- (3) DFLB changes NGINX configuration to not use Instance 2
- (4) DFM restarts the broken components while NGINX proxies requests to Instance 1
- (5) DFM notifies DFLB that the component is back up
- (6) DFLB changes NGINX configuration back to use the 2 instances



The Scenario

Broken Instance

05 Load Balancing

- (1) NGINX sends a notification to LB to inform about a broken instance (Instance 2)
- (2) DFLB changes NGINX configuration to not use Instance 2
- (3) DFLB sends a request to Cloud Provider to restart the instance (Instance 2)
- (4) When everything is back up (instance + all components), DFLB changes NGINX configuration back to use Instance 2